In the Claims:

Please amend the claims as follows:

1. (Currently Amended) A method for maximizing group membership comprising:

updating a connectivity count of each vertex in a graph after removing one vertex from said graph, wherein each vertex represents a single hardware component, and wherein the connectivity count of a vertex is a number of neighbors connected to the vertex;

placing vertices in decreasing order of connectivity based upon said calculated connectivity count of each vertex in said graph;

selecting a vertex with a least sum of connectivity counts of all neighboring vertices from among all vertices having a least connectivity count;

removing said selected vertex from the graph; and

returning a grouping of interconnected vertices forming a clique of completely interconnected vertices ~~in response to said connectivity count of a least connected vertex becoming equal to a number of remaining vertices in the graph~~, wherein each vertex in said grouping is connected to each other vertex in said grouping.

2. (Original) The method of claim 1, further comprising updating said connectivity count for all remaining vertices in said graph following removal of a single vertex from said graph.

3. Cancel

4. Cancel

5. (Previously Presented) The method of claim 1, wherein said vertex is selected from a group consisting of: a computing node, components on a circuit board, division of points in a pattern, and partitions of items.

6. Cancel

7. (Currently Amended) A system to determine a maximum group membership comprising:

a graph with at least two vertices;

a counter to calculate a connectivity count for each vertex in the graph, wherein each vertex represents a single hardware component, wherein the connectivity count of a vertex is a number of neighbors connected to the vertex;

a placement of each vertex in descending order of connectivity based on said calculated connectivity count;

a selection of a vertex with a least sum of connectivity counts of all neighboring vertices from among all vertex with a least connectivity count;

a removal of said selected vertex from the graph; and

a clique of completely interconnected vertices formed ~~in response to the connectivity count of a least connected vertex being equal to a number of remaining vertices in the graph~~, wherein each vertex in the clique is connected to each other vertex in the clique.

8. (Previously Presented)The system of claim 7, further comprising an update of connectivity for each of said vertices subsequent to said removal of a vertex from said graph.

9. (Currently Amended) The system of claim 7, wherein removal of a vertex from said graph with said connectivity count is continuous until the clique of completely interconnected ~~said connectivity count of a least connected vertex is equal to a number of remaining~~ vertices is formed ~~in the graph~~.

10. (Previously Presented) The system of claim 7, wherein said vertex is selected from a group consisting of: a computing node, components on a circuit board, division of points in a pattern, and partitions of items.

11. Cancel

12. (Currently Amended) An article comprising:

a computer-readable recordable data storage medium;

means in the medium for updating a connectivity for each vertex in a graph, wherein each vertex represents a single wherein each vertex represents a single hardware component, and the connectivity count of a vertex is a number of neighbors connected to the vertex;

means in the medium for placing vertices in decreasing order of connectivity based upon said calculated connectivity count of each vertex in said graph;

means in the medium for selecting a vertex with a least sum of connectivity counts of all neighboring vertices from among all vertices having a least connectivity count;

means in the medium for removing said selected vertex from the graph; and

a clique of completely interconnected vertices formed ~~in response to the connectivity count of a least connected vertex being equal to a number of remaining vertices in the graph~~, wherein each vertex in the clique is connected to each other vertex in the clique.

13. Cancel

14. (Previously Presented) The article of claim 12, wherein said means for removing a least connected vertex for removal from a clique in said graph includes comparing a connectivity count of said least connected vertex with a number of remaining vertices in the graph.

15. (Original) The article of claim 12, further comprising means in the medium for updating connectivity for each remaining vertex in said graph subsequent to removal of said least connected vertex.

16. (Previously Presented) The article of claim 12, wherein said vertex is selected from a group consisting of: a computing node, components on a circuit board, division of points in a pattern, and partitions of items.

17. Cancel

18. (Currently Amended) The method of claim 1, wherein the step of removing each selected vertex from the graph is continuous until the clique of completely interconnected vertices is formed ~~connectivity count of a least connected vertex is equal to a number of remaining vertices in the graph~~.

19. (Previously Presented) The method of claim 1, further comprising noting a removed vertex with a connectivity count equaling zero together with all vertices removed in previous iterations which connectivity count at the time of removing was one greater than a connectivity count of a vertex removed in the previous iteration, said noted vertices forming a clique, with the number of vertices in said clique being noted.

20. (Previously Presented) The method of claim 19, further comprising determining a maximum clique in said graph by comparing the number of vertices in said noted cliques.